
BitCom
A Communications
Program
for IBM PC

EVEREX

EVEREX SYSTEMS, INC.
48431 MILMONT DRIVE
FREMONT, CA 94538

BitCom

A Communications Program for IBM PC

User Manual



BIT Software Inc.
P.O. BOX 619
Milpitas, CA 95035

Copyright, BIT Software Inc., 1984, 1985.
All Rights Reserved.

BIT Software, Inc.

Program License Agreement

CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THIS DISKETTE PACKAGE. OPENING THIS DISKETTE PACKAGE INDICATES YOUR ACCEPTANCE OF THE FOLLOWING TERMS AND CONDITIONS. IF YOU ARE NOT IN AGREEMENT, PROMPTLY RETURN THE PACKAGE UNOPENED, AND YOUR MONEY WILL BE REFUNDED.

LICENSE

- a. The program may be used on a single machine.
- b. The program may be copied into any machine readable or printed form for backup or modification purposes in support of your use of the program on the single machine.
- c. You may transfer the program and license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the program, you must either transfer all copies, whether in printed or machine-readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE PROGRAM, OR ANY COPY, MODIFICATIONS OR MERGED PORTION, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE.

IF YOU TRANSFER POSSESSION OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

LIMITATIONS OF REMEDIES

BIT's entire liability and your exclusive remedy shall be:

1. the replacement of any diskette(s) not meeting BIT's "Limited Warranty" and which is returned to BIT or an authorized BIT Software dealer with a copy of your receipt, or
2. if BIT or the dealer is unable to deliver a replacement diskette(s) which is free of defects in materials or workmanship, you may terminate this Agreement by returning the program and your money will be refunded.

IN NO EVENT WILL BIT BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PROGRAM EVEN IF BIT OR AN AUTHORIZED BIT SOFTWARE DEALER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

What is BitCom?

BitCom is a powerful asynchronous communication package for the IBM PC. With BitCom, your IBM PC can talk to the outside world through a Hayes compatible modem, or be directly connected to another computer. It can be used for calling other computers, allowing other computers to call you, or just as a phone list database.

Common uses for BitCom are:

- Electronic Bulletin Board Services.

Supplied on the program diskette are several phone numbers of electronic bulletin board services. You can dial in to exchange messages or get information related to Personal Computers. There is no access fee charged for using the services except for your telephone charge.

- Electronic Banking.

Some banks such as Bank of America provide a personal banking service which allows you to dial in the bank's computer to get information about your accounts, or transfer funds. You should contact your local banking office to get information such as phone number and communications parameters such as speed and parity.

- Personal Dialing Directory.

You can simply use BitCom as a telephone dialer to store your friend's phone numbers or addresses. With a modem supporting Auto-dial (like the Hayes modem), you can dial any number in the list for voice conversation as well as data transfer.

- Sending or Receiving Telex Messages.

You can call up Western Union's EasyLink service to send or receive telex messages. Because BitCom lets you run any DOS programs concurrently, you can easily prepare your message with a word processor and send them through EasyLink.

- Transferring Files or Data With Another Computer.

You can transfer data or files with other computers. BitCom's data capture feature lets you capture any display onto a disk file.

- Terminal Emulator.

BitCom can emulate any terminal by using an emulation file. By setting up an emulation file, you can emulate almost any terminal functions. However, included in the package are 3 popular terminal emulators: VT52, VT100 and IBM 3101.

TABLE OF CONTENTS

1.0	INTRODUCTION	
1.1	Feature Highlights	5
1.2	System Requirements	6
1.3	Getting Started	7
1.4	Reading This Manual	9
2.0	USING BITCOM	
2.1	Starting the BitCom Program	10
2.2	Getting Help	11
2.3	Dialing and Communicating	11
2.4	Filling Out the Blanks	13
2.5	Global Configuration Menu	14
2.6	Phone Number Selection Menu	18
2.7	Setting Up Communications Parameters	20
3.0	COMMON ACTIVITIES	
3.1	Capturing Data	27
3.2	Sending Files	27
3.3	Receiving Files	28
3.4	Answer Mode	29
3.5	Automatic Logon	29
3.6	Terminal Emulation	30
3.7	Avoiding the Menus	31
4.0	Questions and Answers	32
	Appendix A - Script/Action Files	34
	Appendix B - Emulation File Format	46
	Appendix C - VT100 and IBM 3101 Emulator Key Board Assignments	51
	Appendix D - BitCom Error Messages	52
	Appendix E - Script File Error Messages	56
	Appendix F - List of Files on BitCom Diskette	60
	Appendix G - ASCII Code Table	61
	BitCom Release 2.3 Update Notice	62

1.0 INTRODUCTION

1.1 Feature Highlights

- Easy to install and setup.

You can set up your communications parameters and options by simply filling in the forms. Multiple choice selections such as baud rate ensure you that only valid parameters are selected.

When you do have questions, by pressing a key, you can get instant help information which is related to the current context. Over 50 pages of help information are available.

- Menu bypass for batch operation.

Once you set up your communications parameters, you can run BitCom with just a command line from DOS. You don't need to wade through layers of menus to start communications.

- Supports 4 serial ports (COM1, COM2, COM3, COM4), leaving you more ports for other purposes such as a mouse or serial printer.

- Supports 132 columns by 44 rows.

With Everex's Graphic Edge card, BitCom can handle screens in 132 by 44 mode. This is perfect for communicating to mainframe computers which require 132 columns.

- Interrupt driven for high speed communications.

BitCom is interrupt driven. It can handle speeds up to 9600 baud. With built-in buffers and XON/XOFF flow control protocol, BitCom is perfect for heavy duty communication to host mainframe or mini.

- Text or binary file transfer.

You can send or receive text file as well as binary (program) files to or from another computer. BitCom supports XMODEM protocol with binary file transfer.

- Runs any DOS program without disconnecting.

You can run any DOS program without having to disconnect your communications lines. For example, you can use BitCom to log on to Dow Jones services and transfer data onto your disk file. You can then invoke your spreadsheet program to do

analysis while keeping the line connected for more transactions. In addition, you can define a special key to invoke a DOS command while you are still communicating with a host.

- Auto-dial, auto-answer and auto-log on.

With a Hayes compatible modem, you can use BitCom as a dialing directory for things such as personal banking, calling your friends, or dialing an on-line data base. You can store several hundred phone numbers. With each number, you can specify related data such as description, baud rate, and other communications parameters.

- Terminal emulation.

BitCom can emulate most of the popular terminals such as ADM-3A, VT100, and IBM 3101. A Configuration table lets you set up special terminal codes and mappings.

- Tailor make your own communications procedures.

BitCom includes an integrated communication language stored as SCRIPT files or ACTION files. Using its statements, you can build your own log-on procedures or data base retrieval procedures. Script files contain commands to control connection and log on. Action files contain commands to talk to the host after you have connected. There are also commands to assign key strokes to save typing.

1.2 System Requirements:

To run BitCom, you need to have:

1. IBM PC, XT or AT with at least 256K of RAM.
2. DOS 2.0 or later.
3. One 160K or 320K drive.
4. An internal modem card, or an Asynchronous Communications card connected either to an external modem, or to another computer.
5. Any 80 column monochrome or color display.
6. Any printer (optional).

1.3 Getting Started

Before you start using BitCom, you should first back up your BitCom diskette in case you accidentally modify it.

To do so, follow the following procedures:

For fixed disk systems (IBM PC XT, AT).

1. With the diskette drive door open, power up your IBM PC XT.
2. DOS should come up and ask for the DATE and TIME. Simply enter in the time and date and press RETURN.
3. Insert the BitCom diskette in drive A and close the drive door.
4. You may want to create a sub-directory for BitCom. To do so, type:

```
MKDIR BITCOM  
and press ENTER.
```

Then set the current directory to BITCOM by typing:

```
CD BITCOM ←
```

5. Next, copy all files from your BitCom diskette to the fixed disk by typing:

```
COPY A:.* C: ←
```

Now, you should be able to run BitCom from the fixed disk.

For two diskette systems:

1. Insert DOS diskette in drive A.
2. Turn the computer power on.
3. DOS should come up and ask for the DATE and TIME. Simply enter in the time and date and press RETURN.
4. Insert a blank diskette in drive B and type:

```
FORMAT B:/S
```

and press RETURN.

Then strike any key to proceed.

5. After formatting is completed, your diskette in drive B should have DOS on it.

Next, copy all files on your BitCom diskette to your diskette in drive B by following these steps:

Remove the DOS diskette in drive A and insert the BitCom diskette in drive A.

Type:

COPY A:*.* B:

and press ENTER. You should see a list of files being copied onto drive B.

Use the diskette in drive B as your working diskette. Keep your original diskette as master.

Before you start.

Before you start the BitCom program, make sure you check the following items.

1. Your modem is properly installed.

If you are using an external modem, make sure you have the right cable connected to asynchronous communications port 1 or 2.

Make sure that you don't have the same port number in another adaptor card (i.e. multi-function card with a serial port).

Consult your hardware manual for more information.

If you are using an internal modem with CORONA PC, AT&T PC, Leading Edge, EAGLE PC, ZENITH 150/160 PC, you must set the port on your modem to COM2. If you have a COLUMBIA PC, your modem must be set to COM3.

2. Make sure that the DSR signal on your modem is on.

If you are using an internal modem, there is probably a jumper connector to set DSR on. Again, read your modem manual.

3. Make sure that your telephone jack is plugged into the right connector on your modem. Most modems have 2 plugs. The plug marked "LINE" should be connected to the wall to the phone company. The plug marked "PHONE" should be connected to your phone set.
4. Make sure that you can use your phone to dial and answer after you plug in the jacks and before you power up your computer.
5. Make sure you can start DOS. Make sure you have the file: COMMAND.COM on your BitCom diskette or in your fixed disk.
6. Find out the telephone number of the computer you want to dial to. Write down its parameter settings such as:
BAUD RATE, PARITY, DATA BIT, STOP BIT.

You must set up BitCom to match these parameters before you can connect to it.

Now, you should be able to use BitCom.

COM ports (COM1,2,3,4)

Your IBM PC supports 4 serial ports. Most internal modem and serial cards have internal switches that you can set to run as COM1, COM2, COM3, or COM4. This way, you have more serial ports for other equipment. Each serial port should be set up to have its own address. If you have 2 serial ports using the same address, BitCom will NOT work. When you receive BitCom, it is set to run on COM1.

1.4 Reading This Manual

This manual does not attempt to teach you all about the world of communications. That is best left up to the many textbooks on the subject. Instead, the manual attempts to present what BitCom can do (and how you make it do what you want) as briefly as possible.

We recommend you read this manual while viewing the BitCom menu screens. In this way you can try things out when you want, and the text will be much clearer.

Advanced features of BitCom are reserved for the appendixes. You may decide to skip reading of the appendixes if you have learned enough to get what you want done from the previous sections.

2.0 USING BITCOM

2.1 Starting the BitCom Program.

If you are using a fixed disk, simply power up the computer with the diskette drive door open. Make sure you are in the directory where BitCom is stored.

If you are using a two diskette drive system, insert your newly formatted diskette with DOS and BitCom into drive A and power up your computer.

Then type:

BITCOM

and press ENTER.

Now, you will see a menu showing you four options as shown in figure 1.

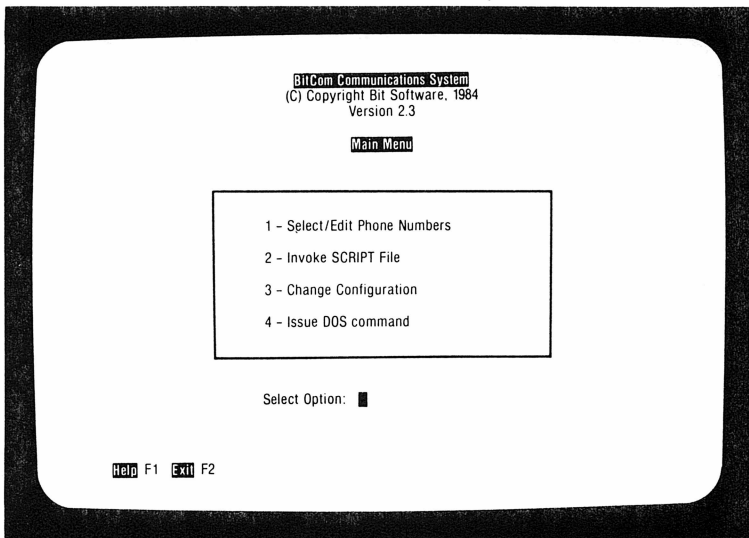


Figure 1. Main Menu

Before you go any further, you should select number 3 to make sure your COM port number is correct.

Press the number 3 and you should see a screen as shown in figure 3.

If your modem or serial port is set to COM 2, you need to change BitCom's port address to COM2. Simply use the down arrow key to move the cursor under PORTS and press the right arrow key to change it to COM2.

While you are in the configuration menu, you may want to change your printer port address (say from LPT1 to LPT2) or other key board assignments.

Section 2.4 and 2.5 describes more detail about each item inside this menu.

When you are done, press F2 to return to the main menu.

Next, you may want to see a list of phone numbers and add your own to the list.

Press the '1' key to see a phone number selection menu as shown in figure 4. This menu will allow you to initiate a call, setup the computer to answer a call, or select a record so you may see/edit the communication parameters.

2.2 Getting Help.

At any time, you can press the F1 key, and you will be shown a short description of the BitCom function keys which have an effect for the current level. You may also press PgDn for more information on that level, or F1 again for a high level index from which you may select help on any BitCom topic.

2.3 Dialing and Communicating.

When you are in the "phone number selection menu", you will see a list of phone numbers and descriptions.

If this is your first time using BitCom and your modem, we suggest you dial into EASYLINK number to make sure your modem and your configuration is working.

Now, move the cursor to a number desired (using the up-arrow and down-arrow keys) and press D to Dial. You will see the program dial the number and wait for connection. If the calling number (host) is connected to a modem and it answers, you can start typing and receiving messages (see figure 2).

To add a number, press the F10 key. A screen as shown in figure 4 will be displayed.

To change an existing number to fit your needs, simply press "S". A screen as shown in figure 4 will be displayed.

Section 2.6 gives you more detail about each item on the screen.

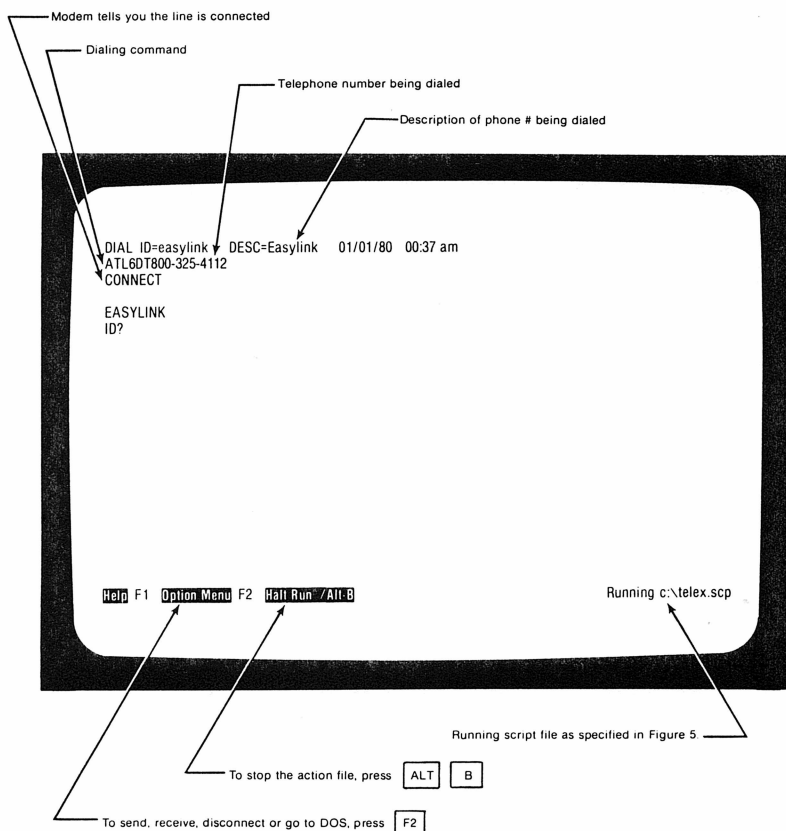


Figure 2. Operational Screen for Using Commands

2.4 Filling Out The Blanks (Fields).

When you are presented with the communications menu or the parameter menu, you will see many blanks. You can simply fill in these blanks or select the appropriate value for the blanks. Each blank area is called a field.

There are four types of fields in each menu as follows:

STRING In this field type, you may enter a string. The right arrow and left arrow keys move the cursor right and left, plus the following keys give the indicated function:

Ins — Turn on/off insert mode. Characters typed will be inserted at the current position.

Del — Delete the character above cursor and shift string left.

BS — Backspace, delete character to left of cursor and shift string left.

Home — Move cursor to 1st position in field.

End — Move cursor one past the last non-blank in field.

For example, the field: Dial Prefix is a string field where you key in the modem dialing command.

SELECT In this field type, the only valid keys to press are left arrow and right arrow, which select the next or previous field value.

SELNUM This is like a SELECT field, except that when the value in the field is all numeric, you may press the keys '1' through '9' and '0' to enter a number.

KEY Here you may press ESC, and then press any special key. The field will be assigned the value of that key.

2.5 Global Configuration Menu

Press the '3' key in the main menu to see/ edit the global configuration parameters (figure 3). These parameters allow you to specify the configuration of your computer system and your modem. You can also change the function key assignment for things like help, confirm, next field, previous field, add record or delete record.

BitCom Communications Configuration			
Modem Controls			
Dial Prefix	:	ATL6DT	
Dial Suffix	:	S0d	
Answer String	:	ATSO=1S0d	
Hangup String	:	+++150ATZS0d\150S0d	
Busy String	:	BUSY	
Connect String	:	CONNECT	
DisConn String	:	NO CARRIER	
Home Directory	:		
Special Keys			
Help	:	F1	Sort Key : F8
Confirm	:	Return	Next Rec : PgDn
Quit	:	F2	Prev Rec : PgUp
Next Field	:	DnArr	
Prev Field	:	UpArr	
Next Sel	:	RiArr	
Prev Sel	:	LiArr	
Add Record	:	F10	
Del Record	:	F9	
Ports			
Comm	:	COM1	
Printer	:	LPT1	
Timing Nulls	:	0	

Help F1 Quit F2 Next Field Prev Field

Figure 3. Global Configuration

FUNCTION KEYS:

- | | |
|------------|---|
| F1 | Show this help page. |
| F2 | Return to Main Menu. |
| UpArrow | Move cursor to previous field. |
| DownArrow | Move cursor to next field. |
| RightArrow | Move cursor right, or select next field value. |
| LeftArrow | Move cursor left, or select previous field value. |

Here you may view or change the parameters which specify the configuration of your computer system and your modem.

We suggest that you read your modem manual and find out the commands such as auto dialing and answer. If you are using a Hayes compatible modem, the parameters should be all set.

The keys down-arrow and up-arrow allow you to move from field to field. The keys left-arrow and right-arrow function differently depending on what type of field you are in.

In the following field descriptions, each field name will be followed by one of the previous field type names (STRING, SELECT, or KEY).

Dial Prefix *STRING*

A string which will be sent to the modem before the Phone Number. For example, ATDT is the dialing command to instruct the modem to dial touch tone. If your modem has the capability to adjust the speaker volume by software, you may want to enter:

ATL7DT to turn the volume to maximum.

ATL0DT to turn off the volume.

To do pulse dialing (rotary dial), enter

ATL6DP or simply ATDP

Dial Suffix *STRING*

A string which will be sent to the modem after the Phone Number. Normally, you would want to send a Return code (hex 0D) to the modem after dialing the number. To enter a hex code, simply enter \$ followed by xx, where xx is a hex digit.

+++\\150ATZ\$0d\\150\$0d

Answer String *STRING*

This string is sent to the modem to enable it to answer the phone. The default command is ATS0=1\$0d. This will ask the modem to answer the phone in one ring. You may want to change the 1 to a larger number so the phone will ring more before answering.

Hangup String *STRING*

This string is sent to the modem to tell it to hang up the phone.

NOTE:

You may specify special characters in the above strings using hexadecimal notation \$xx (e.g. \$0D for Carriage Return), and specify time delays as \\nnn where 'nnn' is the number of hsecs (e.g. \\050 = wait 1/2 sec).

Hayes modems require that you put three plus signs (+++) followed by some time delay and the string "ATZ" to hang up the modem. The default suffix setting in BitCom is:

+++\\150ATZ\$0d\\150\$0d

Busy String *STRING*

This is the string which the modem sends out to mean the dialed number was busy. The default is BUSY. Some modems may not support busy detection.

Connect String *STRING*

This is the string which the modem sends out to mean connection has been established. The default is CONNECT. You should not change this string and the disconnect string unless your modem sends back a different string to mean connection.

DisConn String *STRING*

This is the string which the modem sends out to mean connection has been terminated.

Home Directory *STRING*

This is the name of a directory or disk which contains the Bitcom files. You may locate the Bitcom files *.HLP, COMM.DAT, and *.EMU in a separate directory or disk drive. And by using the DOS 'PATH' command, you may locate the BITCOM.EXE file in a separate directory. However, the COMM.GLO file must always be located in the current directory.

Help *KEY*

The key which places you in the help world whenever you are looking at a BitCom menu (i.e. not in communications).

Confirm *KEY*

The key which confirms you are ready for a given event (e.g. to begin communications or to begin file transfer).

Quit *KEY*

To Quit the current level and return to a previous level menu, or exit to DOS if at the Main Menu.

Next Field *KEY*

To move to the next field when in a menu.

Prev Field *KEY*

To move to the previous field when in a menu.

Next Sel *KEY*

To select the next value in a SELECT field, or move the cursor right in a STRING field.

Prev Sel *KEY*

To select the previous value in a SELECT field, or move the cursor left in a STRING field.

Add Record *KEY*

Key to add a record to the database when in the Selection or Communications Parameters menus.

Del Record *KEY*

Key to delete a record to the database when in the Selection or Communications Parameters menus.

Next Rec *KEY*

Key to display next record in the Communications Parameter menu, or display the next page in the Help or Selection menus.

Prev Rec *KEY*

Key to display previous record in the Communications Parameter menu, or display the previous page in the Help or Selection menus.

Sort Key *KEY*

Key that initiates sorting the database while in the Selection menu.

Comm *SELECT*

The communications port to be used. Usually COM1 or COM2.

Printer *SELECT*

The printer port to be used. Usually LPT1 (a parallel port). To use a serial port as the printer, use the DOS command 'MODE LPTx:=COMy' where 'x' = 1, 2, or 3 and 'y' = 1 or 2.

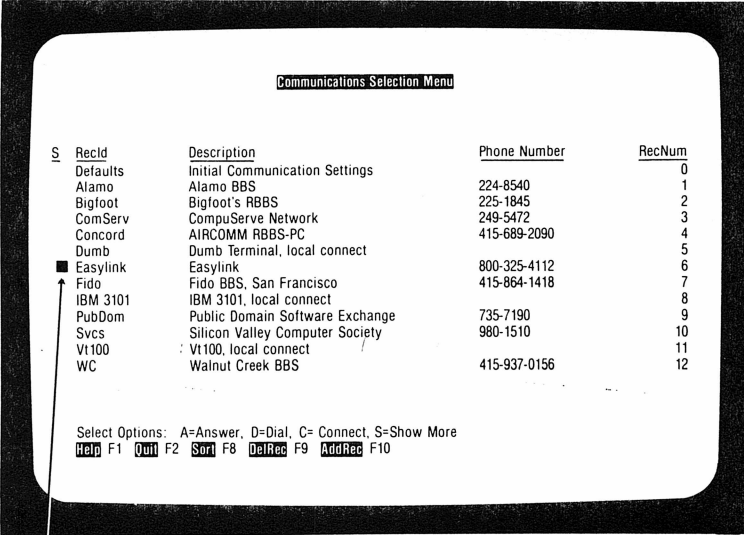
Timing Nulls *KEY*

Number of null characters to be written to the printer after each line. Usually 0 unless your printer is on a serial port.

2.6 Phone Number Selection Menu (figure 4)

Press the '1' key in the main menu to see the Phone Number Selection Menu. You may move the cursor to a desired phone number and press one of the following keys:

- D Dial the number
- A Set computer to answer mode and wait for call
- C Establish immediate connect (i.e. if there is no modem)
- RETURN Dial, Answer, or direct Connect depending on current Mode setting for the number
- S Review/Change phone number, description or communication parameters such as baud rate.
- PgDn Next page of phone number.
- PgUp Previous page.



The screenshot shows a terminal window titled "Communications Selection Menu". It contains a table with four columns: "S", "RecId", "Description", "Phone Number", and "RecNum". A selection pointer, represented by a small square, is positioned to the left of the "Easylink" row. Below the table, there is a line of text: "Select Options: A=Answer, D=Dial, C= Connect, S=Show More". At the bottom, there are function key shortcuts: "Help F1", "Quit F2", "Sort F8", "DelRec F9", and "AddRec F10". A line from the caption points to the selection pointer.

S	RecId	Description	Phone Number	RecNum
	Defaults	Initial Communication Settings		0
	Alamo	Alamo BBS	224-8540	1
	Bigfoot	Bigfoot's RBBS	225-1845	2
	ComServ	CompuServe Network	249-5472	3
	Concord	AIRCOMM RBBS-PC	415-689-2090	4
	Dumb	Dumb Terminal, local connect		5
■	Easylink	Easylink	800-325-4112	6
	Fido	Fido BBS, San Francisco	415-864-1418	7
	IBM 3101	IBM 3101, local connect		8
	PubDom	Public Domain Software Exchange	735-7190	9
	Svcs	Silicon Valley Computer Society	980-1510	10
	Vt100	Vt100, local connect		11
	WC	Walnut Creek BBS	415-937-0156	12

Select Options: A=Answer, D=Dial, C= Connect, S=Show More

Help F1 Quit F2 Sort F8 DelRec F9 AddRec F10

Use 1 or 1 key to move this selection pointer.

Figure 4. Telephone Directory

The Phone Number Database

In the Phone Selection Menu you are actually seeing a high level view of a database of communications information. Each line in the display represents one record of this database. You may have up to 32,000 records in the database, though this would not be

practical for many reasons. However, a database of several hundred numbers is perfectly reasonable if you have a need for that many. Each record in the database has a RecId (a Record Identifier) which may be a maximum of 8 characters in length. The RecId is used when writing "scripts" which describe what numbers to call and when. Each record also has a Description in which a longer description of the record may be placed. And each record may have a Phone Number which is used when you are ready to dial a number. The Phone Selection menu also displays the RecNum (Record Number) so you know where you are in the Phone Number Database.

Your telephone database is stored in a file called: COMM.DAT.

Adding Records

To add a record to the database, press the F10 key. You will be presented with a Communications Parameters display, which is a detailed display of your new record in the database. It will be filled in with the default values from record 0 of the database. With this display you can fill in the RecId, Description, and Phone Number fields, and change any other fields as needed. Note that filling of these fields is purely optional, and is done so you can more easily find the information again later.

Deleting Records

To delete a record in the database, position the cursor at the RecNum you want to delete, and press the F9 key. You will then see a message that asks you if you really want to do this, and if you press the 'Y' key the record will be deleted. Press the 'N' key and the Database will be left unchanged. Note that when you delete a record, the Database will not get any smaller. An empty hole will be left in the database where the record used to be. To make the database smaller, you must 'sort' it.

Sorting Records.

Press the F8 key to sort records in the database. You may sort either by the RecId, the Description, or the Phone Number. When you first press F8 you will see a message asking you to enter an 'I' to sort by Id, a 'D' to sort by Description, or a 'P' to sort by Phone Number. After you answer you will be asked if you want to sort in Ascending ('A') or Descending ('D') order. Once you press the appropriate key the sort will begin. If the database is very large, this could take several minutes. Also, you must have enough space in the current disk drive for two copies of the database, since the sort creates a new version of the database and then deletes the old one and renames the new one (this is much safer in case you have a machine failure during the sort).

Dialing Numbers

To dial a number, position the cursor next to the record you want, and press 'D'. This will set the communications parameters as given in that record, present you with a communications screen, and dial the number.

Answering

If you wish another computer to be able to call your computer, position the cursor next to a record that has the proper communications parameters and press 'A'. You will be presented with a communications screen waiting for a connection.

Direct Connect

If you are directly connected to another computer (i.e. not through a modem), position the cursor next to a record that has the proper communications parameters and press 'C'. You will be presented with a communications screen connected.

Show More/Editing

If you merely wish to "See More" information about a record, or possibly want to change information, position the cursor next to the desired record and press 'S'. This will present you with the Communications Parameters Menu.

Editing Defaults

You may see/change the default communications parameters by placing the cursor at record 0 and pressing 'S'. These defaults are used whenever you add a new record with the F10 key.

2.7 Setting Up Communications Parameters.

Function keys available in this menu:

F1	Show this help page.
F2	Return to Selection Menu.
F9	Delete current record.
F10	Add a new record.
PgDn	Show next record.
PgUp	Show previous record.
UpArr	Move cursor to previous field.
DnArr	Move cursor to next field.
RiArr	Move cursor right, or select next field value.
LeArr	Move cursor left, or select previous field value.
Return	Begin communications using current values.

Communications Parameters screen is shown in figure 5.

BitCom C Communications Parameters			
Record Id	: EasyLink	Last Connect Date: 01/01/80	
Description	: Easylink	Time: 00:17 am	
Phone Number	: 800-325-4112		
Comm Options	Special Keys	Filters	Delays
Baud : 1200	Help : F1	Trace : None	Csend : None
Parity : Even	Escape : F2	Output : Yes	Lsend : None
Data : 7	Show Fkey : F3	Input : Yes	Lchar : 0
Stop : 1	Print : F4	Auto LF : No	BreakT : 40
Echo : Yes	Snapshot : F5	Exp Tab : No	ReDial : 0
Mode : Call	Break : Alt-B	Up Case : No	
Ignore : None		BlankLn : No	
Other			
Capture File	: C:\TELEX.CAP	Capture Mode : Append	
Autolog File	: c:\qictelex.scp		
Emulation	: None		
Notes			

Help F1 Quit F2 Next Fld Prev Fld Begin Comm Rec 6/ 12

Figure 5. Setting up Parameters for each Communications Line

In this screen you may give the multitude of communications parameters that must be set for the wide variety of over complicated systems in the world. The keys UpArr and DnArr allow you to move from field to field. The keys LeArr and RiArr function differently depending on what type of field you are in. There are four types of fields as follows:

In the following field descriptions, each field name will be followed by one of the previous field type names (STRING, SELECT, SELNUM, or KEY).

Record Id *STRING*

A record identifier to be used when writing "script" files. The maximum length is 8 characters. The id should be unique for all records in the database. The reason for only 8 characters is to minimize processing time when running a "script".

You would also need this record id if you want to run BitCom directly from DOS and bypassing menus.

Description *STRING*

A longer description of the record, usually more useful to human than the 8 character ID.

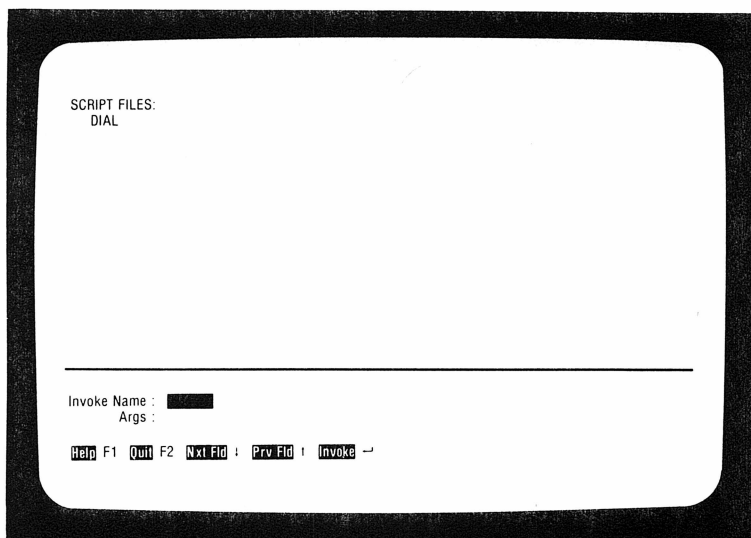


Figure 6. Invoking a Script File

Phone Number *STRING*

The number to be dialed. Up to 40 characters may be given so there is room for long distance prefixes, area code, number, and modem commands. Note that when the number is dialed, the global "Dial Prefix" and "Dial Suffix" strings will be sent before and after the number.

A phone number may consist of a numeric digit, a space, commas (,) or minus sign (-). To delay dialing the next digit, insert a comma for each one second delay. (Each comma will delay an amount of time specified by Hayes command. See Hayes command description for detail).

For example, the number 9,,,408-555-1212 tells the modem to dial 9 then wait 3 seconds, and dial the number 4085551212. Note that the minus signs are ignored.

You may also use the modem as a telephone dialer. This means that you tell the modem to dial the phone but do not connect as a data line. To do so, you must connect a phone set to the second plug of your modem. Then you set up the phone number in the directory. The phone number must be followed by the letters: ;H0 to tell the modem to hang up after dialing. You pick up the phone and press D to dial the

number. After the modem dials, it will hang up leaving your phone set connected for voice conversation. For example, the phone number 408-555-1212;H0 will tell the modem to dial and hang up.

Last Connect Date and Time

These are not input fields, but are here to show you when you last made a successful connection to the current number.

Baud *SELECT*

The Baud rate you wish to communicate at, normally 300 or 1200 for modems.

NOTE:

You must find out the BAUD RATE, PARITY, STOP BIT and DATA bit of the phone number you dial into and match the same settings here. Otherwise your modem will not connect.

Parity *SELECT*

The parity checking method to be used. Possible values are None, Even, Odd, Mark, or Space.

Data *SELECT*

The number of bits to use for Data. Only valid values are 7 or 8. Note that if XMODEM file sending/receiving is to be done, 8 is required.

Stop *SELECT*

The number of stop bits to use. Only valid values are 1 or 2.

Echo *SELECT*

Either 'Yes' which tells BitCom to display each character you type, or 'No' which implies that the computer you are talking to will do the Echo (i.e. it will send back each character as it receives it).

This option is also known as "FULL/HALF DUPLEX".

If you see double the character you type (when you are connected), you should set ECHO to NO.

You may change this option while in connection with the host.

Mode *SELECT*

Either 'Call', 'Ansr' or 'Drct'. If 'Call', when you press Return, the number will be dialed. If 'Ansr', the number is not dialed but the

modem "Answer String" is sent. If 'Drct', a direct connection is assumed (nothing is sent).

Ignore *SELECT*

Which communication signals you want to ignore: DSR, CTS or both. If you are connected with a cable to a host computer, your host computer may not provide DSR, CTS or both. In this case, you need to ignore these signals in order for BitCom to work. DSR tells your PC if the data set is ready while CTS tells your PC if it is clear to send.

Help *KEY*

The key which places you in the help world while you are in communications. This key need not necessarily be the same as the usual help key (F1).

Escape *KEY*

The key which presents you with the options menu, whereby you can initiate sending and receiving files, change communications parameters, issue DOS commands, and hangup the phone while communicating.

Print *KEY*

The key which turns on/off sending a copy of all data which goes to the screen to the printer.

Break *KEY*

The key which will send a "Break" signal to the other computer. Note that BitCom (unlike most communications systems) allows you to define the real "Break" key (Ctl-Scroll Lock) for this function.

Snapshot *KEY*

This key will send a copy of the screen (80 columns by 25 lines) to a file or the printer. When you first press this key in a BitCom session, you will be prompted for a filename. Thereafter, when you press it, you will be prompted, but you may merely press F4. In all cases the screen will be appended to the filename you specify.

Trace *SELECT*

This field may be 'None' for no trace, or 'Ascii' or 'Hex'. If 'Ascii', special control codes received are echoed as an @ (at-sign) followed by the name of that code (e.g. @Ctl-B). If 'Hex', special codes are echoed as a \$ (dollar) followed by the hexadecimal version of that key (e.g. \$1B).

Output *SELECT*

If 'Yes', line feed characters are not sent if sending a file in Ascii mode.

Input *SELECT*

If 'Yes', input from the other system will be 'cleaned up' before being displayed. Special control codes are stripped, and the high bit is also removed.

This is also call "stripping characters". This option will have no effect on XMODEM file transfer.

If you see graphics character appearing on your screen, you may want to set INPUT to yes to strip out the high order bit. (Your host computer may be only using 7 bit data.)

Auto LF *SELECT*

If 'Yes', when a carriage return (CR) is received from the host, a line feed (LF) will be written to the display also. Alternatively, when a LF is received, a CR will be written to the display also.

Exp Tab *SELECT*

If 'Yes', tab characters will be expanded to the appropriate number of blanks when sending a file in Ascii mode.

Up Case *SELECT*

If 'Yes', all characters sent to the host will be converted to upper case.

BlankLn *SELECT*

If 'Yes', lines consisting of nothing but blanks will be sent as a single blank character when sending a file in Ascii mode. This is provided because some systems treat a totally blank line as end-of-file.

Csend *SELNUM*

The type of delay to take place after sending each character in Ascii mode. 'None' means no delay, 'Echo' means wait for the host to send a character back before sending the next, or you may enter a number which specifies the number of hundredths of seconds to wait after each character.

Lsend *SELNUM*

The type of delay to take place after sending each line in Ascii mode. 'None' means no delay, 'Echo' means wait for the host to send a line feed (LF) character, 'Manual' means wait until you press a key, or you may enter a number which specifies the number of hundredths of seconds to wait after each line.

You may also give an arbitrary character to wait for from the host during ASCII file transmittal. For example, if your host computer sends you a question mark (?) as a prompt so that you can enter a line of text, you can set LSend to ? and send a whole file over.

BreakT *STRING*

Specifies the length of the break signal to be sent in hundredths of a second. Usually 40/100 of a second is sufficient.

ReDial *STRING*

Specifies the number of seconds to wait before re-dialing the current number. If the number is 0, no re-dial will be done.

Capture File *STRING*

The name of a file to which all output to the screen is to be written once you have begun to establish communications (i.e. after you press the Return key). The file name may be a full pathname up to 40 characters long including drive name.

For example, you can capture a file to drive C, directory \telex\msg by using file name: C:\TELEX\MSG.

Or you can capture a file to drive B by using a name:

B:TELEX.MSG

Capture Mode *SELECT*

Specifies the mode of the capture file. 'Append' means each time you begin communications the file will be appended to. 'Replace' means the file will be erased (if it exists) and re-created each time.

Autolog File *STRING*

The name of a file containing startup commands. This file usually contains function key definitions, and the sign-on sequence for the current number. The file is invoked immediately after a connection is established. File name may contain drive name as well as path name.

Emulation *SELECT*

The type of terminal emulation to be performed. If "None", no emulation will be done. Otherwise it may be the name of a .EMU file which contains a description of the terminal characteristics.

Notes *STRING*

This is two lines in which you may enter any text you like pertaining to this record.

3.0 COMMON ACTIVITIES

3.1 Capturing Data

BitCom provides you with several ways of saving your incoming display or messages onto a file or to your printer.

- 1) Take snapshot of current screen (F5).

This feature lets you save the currently displayed screen (25 lines by 80 columns) onto a file or to your printer.

When you first press the Snap key (F5), BitCom will ask you for a file name to save your screen. Thereafter, each time you press the same key again, it will show you the same file name. Simply press ENTER to save it. If the file already exists, the screen will be 'appended' to the end of the file.

To capture your screen to the printer, either simply press the Shift-PrtSc key on your keyboard, or reply 'LPT1' for a filename in response to the Snap key prompt.

- 2) File capture (given on the Parameter menu).

The file capture feature lets you save all incoming characters onto a text file. Before you can use this feature, you must specify a capture file name in the Parameter menu. You also must specify if you want that file 'Appended' to or 'Replaced' at the beginning of each communications session.

After you specify a file name for file capture, all incoming characters will be automatically written to the file. To stop capturing, either end the session (disconnect) or go to the Parameter Menu by pressing F2 and choose the Parameter menu. Then specify the Capture File name as 'None'.

The file capture feature is useful for setting up auto logon files or to record your communications session.

- 3) Capture to printer with Print key (F4).

This feature begins printing all characters displayed on the screen to the printer until you press the key again.

3.2 Sending Files

You can send any DOS files in TEXT mode (ASCII mode) or in XMODEM mode. In ASCII mode, your entire file is sent to the host without any error detection capability. The host computer will simply treat it as if it were typed in by you. If your host

computer does not have XMODEM capability, this is the only way you can send data to the host.

If your host computer is using a program with the XMODEM feature, you should send your files in XMODEM mode. XMODEM provides error detection capability so if there is any error in the transmission, it will try again.

How you send a file to the host computer depends on the requirements of your host computer. For example, Western Union's Easylink system requires that each command starts with the character slash (/). So if you created a file with your word processor and the first character in that file is a slash, you may simply send the whole file over to Easylink without typing any commands.

In general, to start sending a file, you should first connect to the host computer. You instruct the other end to get ready to receive a file. Consult your host computer's manual for receiving a file from a modem. Most computers use a COPY command to receive a file from a terminal or modem. If the other end is also using BitCom, the person at the other computer can do this by pressing F2 and selecting the RECEIVE option. You then press F2 to get the option menu, and choose SEND.

You will be asked for the file name to send and what mode (ASCII or XMODEM) to use. Fill in the filename. If the mode is not correct, press 'Down Arrow' and use the left and right arrow keys to select what mode to use. Then press RETURN. At this time BitCom will show you the file size, and tell you how long it will take to send the file. Press RETURN to begin sending, or F2 to cancel the send.

You may cancel the send at any time by pressing the F2 key.

3.3 Receiving Files

You can receive a file in TEXT (ASCII) mode or in XMODEM mode. In text mode, there is no error checking. In XMODEM mode, there will be error checking but the sending computer must be sending in XMODEM format. The procedure is similar to sending files. You ask the other end to wait a while for you to set up the file name and mode. Then you press F2 and choose the RECEIVE function. You fill in the blanks and press RETURN. BitCom will wait for the other end to send and save the data. You may cancel the receive at any time by pressing the F2 key.

3.4 Answer Mode

You can use BitCom to tell your modem to answer the phone. Your modem must have the auto answer feature. To use auto answer, you must set up parameters in the Communication Parameter menu. You can use any phone number (any record) on your phone list. You fill in the blanks for baud rate, stop bit, parity and other information that you want to use when the phone is answered. You then return to the phone list and press A for answer.

When the phone rings, your modem should answer. If the caller is calling with a modem, you should be connected after a few seconds.

Again, make sure that the calling modem has set up the same communications parameter (BAUD RATE, STOP, DATA BITS, PARITY).

3.5 Automatic Logon

If you call a number frequently, you will probably want to setup an automatic logon file so you don't have to always remember the correct logon sequence. You do this with BitCom 'ACTION' files. Specify the name of a file in the 'Autolog File' portion of the Communications Parameters menu. This file will be run whenever you 'Connect' to the given number.

To create your own auto-log file (action file), use any word processor capable of creating text files (such as EDLIN, PE). If you use WORDSTAR or any program that includes formatting codes inside a file, then you have to use Non-Document mode (or ASCII mode). Your action file simply contains lines of instructions or statements to tell BitCom how to do auto-log on.

Included on your program diskette are several ACTION files (which have file extensions of .ACT). Refer to Appendix A for a complete description of the BitCom ACTION/SCRIPT file language.

As a simple example, to send a RETURN (CR or hex 0D) to the host computer, simply create a line:

```
"$0d"
```

To send a line feed or new line (hex 0A), create a line with:

```
"$0a"
```

To send a string of characters followed by a RETURN and Line Feed:

```
"HELLO$0d$0a"
```

In fact, you can send any combination of ASCII codes to the host by using the dollar sign \$ followed by a hex number.

Any characters between an open and close square bracket [] will be treated as comments and ignored by BitCom auto-log function.

Following is a sample autolog file to logon to the CompuServe service:

```
[ Autologon file for CompuServe Network ]
twait(1, "sec")      [ wait for 1 second ]
"$03"               [ send Ctl-C character hex 03 ]
cwait("ID:")        [ wait until 'ID:' string received ]
"xxxxx,yyy$0d"      [ send login ID followed by carriage return
                    hex 0d] [ return ]
cwait("word:")       [ wait for pass'word:' prompt ]
"xxxxx/yyyyyy$0d"   [ send password followed by carriage
                    return ]
invoke handykey      [ invoke another file which defines] [ some
                    keys ]
```

The general sequence as given above can be used for most services (i.e.wait for a string, send a response). The file 'HANDYKEY.ACT' defines some useful keys:

```
[ This file defines some handy function keys ]
@alt-t = '[ Display time ] typecr("Current time is @cdate
@ctime")'
@alt-h = '[ Hangup phone ] hangup'
@alt-d = '[ Give DOS cmd ]
@a = prompt("Enter DOS command: "); dos(@a)'
@alt-c = '[ Clear Screen ] clear'
```

You can also create more complicated ACTION files to prompt for information, and operate differently depending on keys pressed or words received from the connected computer.

3.6 Terminal Emulation

If you are communicating with a computer that requires a special terminal type, you may need the terminal emulation feature. This feature lets your PC emulate other terminals such as the DEC VT100 or IBM 3101.

To use terminal emulation, you must set up an Emulation file which contains information about the terminal characteristics. Included with BitCom are 3 terminal emulation files:

VT100A for DEC VT100 with ANSI support

VT100B for DEC VT100 with ANSI and VT52 support

IBM3101 for IBM 3101 terminal.

To set up emulation, you need to go to the Communications Parameter menu for the phone number you want. Under the field "Emulation", fill in the emulation file name. For example, fill in VT100A for DEC VT100. After you have connected to the line, the emulation function will be automatically invoked. Your key board will simulate the appropriate terminals. For details of setting up an emulation file, refer to appendix B.

3.7 Avoiding the Menus

If you know the 'Record Id' of a communications record you want to use, you can bypass the usual menus and prompts and begin communications directly by giving the following command in response to the DOS prompt:

A> BITCOM recid

For example, type:

BITCOM EASYLINK

and press ENTER to start BitCom, dial Easylink and invoke any script file.

You can even set up a batch file to dial and communicate with different computers. For example, you want to dial into Easylink, CompuServ and then Dow Jones. You may want to create a batch file (say DIAL.BAT) which contains the following lines:

BITCOM EASYLINK

BITCOM COMPUSER

BITCOM DOWJONES

4.0 QUESTIONS AND ANSWERS

1. Q. I put the BitCom diskette in drive A and press BITCOM. It seems like the drive is finished reading the program but the computer hangs up. Why?

- A. Make sure that your computer has at least 192K of available memory to run BitCom. Insert the DOS disk in drive A and type: CHKDSK to find out how much memory is available.

Next, check to make sure that the diskette containing BitCom has the correct BITCOM.EXE file. Compare your disk with the original.

2. Q. I started BitCom, dialed a number and got a message saying that COM1 is not functional or NO DSR Signal on COM1. Why?

- A. If you are using an internal modem, make sure that your computer does not have another serial card. If you are not sure, switch the modem to COM2 and set BitCom to COM2 and try again. You may be trying to access another serial card while you are expecting to access the modem which is on the same address.

If you are using an external modem, make sure all cables are securely connected and the modem has the DSR light on.

If your modem or hard wire connection does not have the DSR signal, you must tell BitCom to ignore DSR by going into the Communications parameter menu and IGNORE to DSR (figure 5). Refer to Section 1.3 "Before You Start" for more information.

3. Q. I have everything set up and BitCom dials my host computer. The host computer answers and the dial tone changes into a high pitch tone indicating that their modem is ready. However, I don't see the CONNECT message from BitCom. It just hangs there. Why?

- A. You probably did not set up the correct baud rate, parity, stop bit, or data bit. All these parameters in BitCom have to match the computer that you are connected to or else it won't work. Also make sure your telephone cord is plugged into the right plug.

4. Q. I got connected to my host computer and started typing. Everything I typed appears twice on the screen. Why?
A. You need to set Echo to "NO". Refer to figure 5.
5. Q. I got connected to my host computer but I see graphics characters (line graphics or foreign characters) on my screen. What should I do?
A. You need to go to the communications parameter menu (figure 5) by pressing F2 and change the "Input filter" to YES to strip out the high order bit.
6. Q. How do I use my modem as a telephone dialer?
A. Refer to section 2.7 under PHONE NUMBER.
7. Q. How do I do pulse dialing?
A. Refer to section 2.5 under Dialing Prefix.
8. Q. How do I manually control or dial the modem with BitCom?
A. First set up a record just like any other records except this one will have no phone number and is for direct connection. Then, when the phone directory shows up, press D and you should be in the Communications screen (figure 2). At this point, you can type any Hayes commands to test the modem or send commands to the modem.
9. Q. How do I set up for automatic log-on to my host computer?
A. You use any word processor to create an action file and put the file name under "Autolog File" as shown in figure 5. For more information, refer to section 3.5 and appendix A.
10. Q. How do I set up pulse dialing (rotary dial)?
A. You change the dialing prefix in your Communications Configuration Menu as shown in figure 3. Refer to section 2.5 for more details.
11. Q. I use a PBX system which requires me to dial 9, and wait for the dial tone before dialing the phone number. How do I tell BitCom to do that?
A. You insert commas between the digit 9 and your phone number when you set up your database. Refer to section 2.7 for more details.

Appendix A – Script/Action Files

A.1 Overview

This appendix presents the language rules (i.e. syntax and semantics) and the various procedures and functions available in the BitCom control language.

A.2 Introduction

There are two kinds of files in which language controls may be given:

- 1) **ACTION** files: These have a file extension of .ACT and may contain function key definitions and commands to control automatic signon or other common key sequences. ACTION files are invoked to perform tasks after you have made a connection.
- 2) **SCRIPT** files : These have a file extension of .SCP and may contain commands to control when to call a certain number, how many times to retry, and what to do if the connection is not (or is) established. SCRIPT files are used to actually make a connection.

These files are created with any editor capable of creating standard "text" files (e.g. PE or EDLIN).

BitCom lines consist of assignment statements, conditional statements, branch statements, and function calls. Comments may be placed anywhere in a file by surrounding the comment with opening and closing curly brackets '[' and ']'. However, comments must begin and end on the same single line. Also, alphabetic case is ignored at all times (except for strings being sent to the host or typed at the terminal). Thus 'Bell' = 'BELL' = 'bell'. Before we get into how to write a BitCom statement, we must define the variables.

Normally, in your action file, you want to put one statement on a single line. however, you may want to put multiple statements on one line by separating each statement with a semicolon (;).

A.2.1 Variables

There are several types of variables which can be used at various places in the BitCom language. "Scratch" variables may be assigned a value for use later in a command file. "Argument" variables contain the values of any arguments passed to a command file. "Key" variables allow you to assign values to, and

invoke special keys. And "Builtin" variables allow you to set and query various BitCom values.

All variables begin with the special character '@' followed by the name of the variable. Note that if you really want to include the character @ in a command file, you must do so with two @ characters together (i.e. @@).

Scratch Variables

There are 26 scratch variables to contain work values. These variables are named @A through @Z.

Scratch variables are useful for saving values and resetting the value later in the file.

Argument Variables

When a SCRIPT or ACTION file is invoked, arguments may be passed to it. The argument may be referred to in the file with the names @1 through @9 which represent arguments 1 through 9. Thus a maximum of 9 arguments may be passed. You may NOT assign values to argument variables.

Key Variables

Key variables are the names of the special keys on the keyboard. Every special key has a unique name, and strings may be assigned to the keys or the keys may be invoked as if you pressed them at the keyboard.

The valid key variables are:

@F1	@Home	@Alt-A	@Ctl-A	@Alt-F1
@F2	@End	@Alt-B	@Ctl-B	@Alt-F2
@F3	@PgUp	@Alt-C	@Break	@Alt-F3
@F4	@PgDn	@Alt-D	@Ctl-D	@Alt-F4
@F5	@Del	@Alt-E	@Ctl-E	@Alt-F5
@F6	@Ins	@Alt-F	@Ctl-F	@Alt-F6
@F7	@Esc	@Alt-G	@Ctl-G	@Alt-F7
@F8	@LfArr	@Alt-H	@BS	@Alt-F8
@F9	@RiArr	@Alt-I	@Rtab	@Alt-F9
@F10	@DnArr	@Alt-J	@Lfeed	@Alt-F10
@F1	@UpArr	@Alt-K	@Ctl-K	@Ctl-F1
@Sft-F2	@Ltab	@Alt-L	@Ctl-L	@Ctl-F2
@Sft-F3	@CtlHome	@Alt-M	@Return	@Ctl-F3
@Sft-F4	@CtlEnd	@Alt-N	@Ctl-N	@Ctl-F4
@Sft-F5	@CtlPgUp	@Alt-O	@Ctl-O	@Ctl-F5
@Sft-F6	@CtlPgDn	@Alt-P	---	@Ctl-F6
@Sft-F7	@CtlLeft	@Alt-Q	@Ctl-Q	@Ctl-F7
@Sft-F8	@CtlRight	@Alt-R	@Ctl-R	@Ctl-F8

@Sft-F9	@Alt-S	@Ctl-S	@Ctl-F9
@Sft-F10	@Alt-T	@Ctl-T	@Ctl-F10
@Alt-U	@Ctl-U		
@Alt-V	@Ctl-V		
@Alt-W	@Ctl-W		
@Alt-X	@Ctl-X		
@Alt-Y	@Ctl-Y		
@Alt-Z	@Ctl-Z		

To assign a value to a key, you give a line like:

```
@Alt-S = ""a string"";
```

You may also invoke the key (as if you pressed it at the keyboard) by giving the name of the key as a statement by itself, like:

```
@Return;
```

Builtin Variables

There are several other variables defined which allow you to set and/or query certain information about a BitCom session. These variables allow you to see and set the various parameters in the BitCom Parameters menu without entering the menu. The variable names are:

<i>Name</i>	<i>Variable</i>	<i>Type Contents</i>
@Autofil	String	Name of automatic logon file
@Autolf	Number	Auto linefeed filter, =0 off, =1 on
@Baud	Number	Current baud rate
@Bell	Number	Current bell setting, =0 off, =1 shortbeep, =2 long beep, =3 long warble
@Blankln	Number	Blank line filter, =0 off, =1 on
@Breakt	Number	Number of hsecs break should last
@Capmode	Number	Capture mode, =0 Replace, =1 Append
@Capname	String	Name of capture file
@Cdate	String	Current date in format "mm/dd/yy"
@Col	Number	Current cursor column number
@Conn	Number	=0 if not connected, =1 if connected, =2 if dialed line is busy

@Crec	Number	Current record number
@Csend	Number	Character delay, -1 wait for echo, 0 none, or number of hsecs to wait
@Ctime	String	Current time in format "hh:mm am pm"
@Data	Number	Number of data bits
@Desc	String	Current record description.
@Echo	Number	=0 No echo, =1 Echo on
@Emulate	Number	Emulation mode, =0 none, =1 3101, =2 VT52
@Exptab	Number	Expand tabs filter, =0 off, =1 on
@Input	Number	Input filter, =0 off, =1 on
@Ldate	String	Last connect date in format "mm/dd/yy"
@Lsend	Number	Line delay, -2 manual, -1 wait for LF, 0 none, or number of hsecs to wait
@Ltime	String	Last connect time in format "hh:mm am pm"
@Mcol	Number	Maximum number of columns on current screen
@Mode	Number	=0 Answer, =1 Dial
@Mrow	Number	Maximum number of rows on current screen
@Note1	String	First line of notes
@Note2	String	Second line of notes
@Output	Number	Output filter, =0 off, =1 on
@Parity	String	"N" None, "E" Even, "O" Odd, "M" Mark,
@Phone	String	Current phone number.
@Recid	String	Current record id.
@Redial	Number	Number of seconds between redials
@Row	Number	Current cursor row number
@Stop	Number	Number of stop bits
@Trace	Number	=0 Trace off, =1 Hex, =2 Ascii
@Uppcase	Number	Upper case filter, =0 off, =1 on

A.2.2 Constants

Constants can be used to compare with and set variables. Constants can be either numbers or strings.

Numbers can be given as simple integers only, optionally preceded with a sign character. There are three types of strings:

those enclosed in double quotes ("), those enclosed in single quotes ('), and those enclosed in no quotes at all.

Strings in double quotes (called d-strings) must be preceded and followed by a double quote ("). To specify a string containing a ", specify two in a row. Strings in single quotes (called s-strings) must be preceded and followed by a single quote ('). Again, to specify an s-string containing a ', specify two in a row.

With a d-string, if a variable name is found in the string (e.g. @1) the variable will be substituted with its value. If you wish to immediately follow a variable with a non-blank character, you must follow the name of the variable with a period. The period will be stripped off the resulting string. Thus, if @a has been assigned the value "def", then:

"@a" is "def"

"abc@a" is "abcdef"

"abc@a.ghi" is "abcdefghi"

Note that if a d-string contains a variable name for which no value has been defined, the variable will be substituted with the null string. No error message will be displayed. To include an @ in a d-string, specify two @'s in a row (e.g. @@).

d-strings may also contain the hexadecimal representation of a character. This is often useful to send special characters to the host or search for a special character. Specify a hexadecimal character as "\$xx" where the x's must be a character in the range of 0-F (e.g. the line feed (LF) character can be specified as "\$0A"). As usual, specify 2 \$'s in a row to actually include a \$ in a string.

With an s-string, no variable substitution will take place. This is useful for assigning strings to function keys for later evaluation.

Examples of valid constants are:

1----200-----2

"This is a d-string"

"This is a d-string with a double quote ("") in it"

'This is an s-string with a double quote (") and single quote (')'

Lastly, if you wish to give a string which has no blanks in it, you may omit the quotes (i.e. specifying "ABC" is the same as ABC).

A.3 Statements

A statement can either be a variable statement, an assignment statement, a procedure invocation, or a control statement. Statements may be separated with a semicolon (;) to get more than one statement on a line.

A.3.1 Variable Statements

A variable statement consists of one of the following:

1. The name of a variable (e.g. @1 or @a)
2. A constant string or numeric value
3. The name of a key variable

For cases 1. and 2., the resulting string of the variable or constant is SENT to the communications port as though you had typed the string at the keyboard. If the variable contained a numeric value, the string representing the number is sent (i.e. 200 = "200").

If the name of a key variable (case 3.) is given, if an ACTION string has been assigned to that key then the action will be done. Otherwise the value of the key will be sent to the communications port as though you had pressed that key at the keyboard. Note however that the special keys (e.g. Help, Confirm, etc.) cannot be invoked this way. For Example:

```
"logid";@return;    [ sends "logid" followed by carriage ] [
                    return ]
@a;                 [ sends keystrokes defined in string
                    assigned ]
                    [ to @a variable ]
@1;                 [ sends string in argument one ]
```

A.3.2 Assignment Statement

The assignment statement looks as follows:

variable = value;

The variable may be any valid @ variable described previously. The value may be either a Number, d-string, s-string, other variable name, or function name. Valid assignment statements are:

Baud = 1200;

@Note1 = "We tried to connect at @Cdate";

@Alt-s = 'sendfile(prompt("Send File Name? "), "Xmodem")'

Note that the last example defines the @alt-s key to prompt for a filename, and begin sending that file.

A.3.3 Functions

There are many BitCom functions which may be invoked to perform a variety of tasks. These functions are invoked by giving the function name. Any arguments the function requires must be enclosed in left and right parentheses and separated by commas. The functions always return a value which is either the result of the function or a return code indicating the success or

failure of the task. You may choose to ignore the function result by not assigning it to anything. Thus,

`getchw;`

invokes the function 'getchw' (which will not return until a character is received on the communications line) and ignores the result, while

`@a = getchw;`

reads the next character from the communications line and assigns the value to the @a variable.

Connection Functions:

The following functions may only be used in SCRIPT files, and are used to establish a connection with another computer.

select(recid)

This function "select"s the given recid (the 8-character name given when you define communications parameters). By selecting the record, you are merely making it the "current record". You may then query or change any of the record values using the builtin variables (e.g. @Baud, @Phone, etc.). Note also that changing the record values will cause the disk record to be changed, just as if you had changed the values from the Communications Parameter Menu. Always returns 0. Running of the SCRIPT file is cancelled if the recid was not found.

docomm

This function begins communications, using the current value of @Mode. If @Mode = 0, the computer is set up ready to answer the phone. If @Mode = 1, the current number in @Phone is dialed. If @Mode = 2, it is assumed the port is directly connected (i.e. there is no modem). The function always returns 0 immediately. You must have used the 'select' function previously.

In the case of @Mode = 1 (dial), you must loop test for @Conn=1 if you wish to wait until a connection is made.

Port Input-Output Functions:

The following functions perform I/O to the communications port.

getch

This routine reads a character from the port. If none is available, it returns -1. Otherwise it returns the character read.

getchw

Like *getch*, but waits for the next character if none is currently available.

hangup

This function transmits the hangup string. It returns 0 normally, -1 if the currently defined hangup string (@hangup) is null.

sendfile(filename, mode)

This function initiates a file send. The argument "filename" is a variable or constant string with the name of the file to send. "mode" must be the string "xmodem" or "ascii" and defines the method used to send the file. This function returns 0 after file is sent, or -1 if the sendfile was not found.

recvfile(filename, mode [,eof])

Just like 'sendfile' but the file is received. If the mode is "ascii", you may also give the string which will be taken to mean the 'end-of-file' has been reached. If you omit this argument, the character Ctl-Z (Hex 1A) will be taken as end-of-file. Ctl-Z is typically the EOF character for text files in CP/M and MS-DOS systems.

Display I/O Functions

The following functions are used to display text and read keyboard input. This is useful mostly to prompt for input.

type(string)

This function types the given string at the display. No carriage return is done after the string. Function always returns 0.

typecr(string)

This function also types the given string, but ends the string with a carriage return. Function always returns 0.

bell

This function rings the current alarm (depending on the current Bell setting). This function always returns 0.

prompt(string)

This function types the given string (with no carriage return), and reads from the keyboard until the carriage return key is pressed. The string typed will be returned.

message(string)

This function displays the given string in the message-area (bottom right corner) of the string. This function always returns 0.

poscur(col, row)

This function sets the current column and row of the cursor. The valid ranges are col=1 to 80, and row=1 to 24. The return is always 0.

clear

This function clears the display screen. The return is always 0.

Event Wait Function

The following functions wait until a given event occurs before returning.

wait(hour, min)

Wait until the given hour and minute occurs. The hour must be given in 24 hour clock format. The wait will occur until the next occurrence of the given time. It always returns 0. Note that while you are waiting, any characters received from the communications port will be passed on to the display.

cwait(string)

Wait for the given string to be received on the communications line. The string may be 1 character long to wait for a single character. To wait for a line feed/carriage return combination, give the string "\$0D\$0A". It always returns 0.

wait(time)

Wait until no characters are received on the communications line for 'time' seconds. Returns the last word received. This is useful to determine when the host is prompting you for some input. If you are unsure what the prompt will be, you can check the last word in the prompt and respond accordingly.

twait(time, unit)

Waits for the given amount of time, depending on the value of 'unit'. Unit may be "hsec" (hundredths of seconds), "sec" (seconds), "min" (minutes), or "hour" (hours). It always returns 0. Note that while you are waiting, any characters received from the communications port will be passed on to the display.

Miscellaneous Functions

expr(expression)

This function evaluates the given expression and returns the value. The expression consists of constants, variables, or function calls separated by operators. Valid operators are '+' for addition, '-' for subtraction, '*' for multiply, and '/' for divide. Multiply and divide are done before any addition or subtraction in the expression, but you may control the order of calculation using parentheses. For example, if the variable @a contains the value 200, then:

```
expr( a+1) = 201
```

```
expr( a+2*6) = 212
```

```
expr(( a+2)*6) = 1212
```

dos(string)

This function issues the command as a DOS command. The return is always 0.

trace(n)

This function enables/disables tracing of the running action/script file. Trace(0) disables tracing (the default value). Trace(1) begins tracing. When trace is on, all input lines will be displayed as they are read, and function returns and assignment results will be displayed as they occur. Trace always returns 0.

A.3.4 Control Statements

Control statements are used to control the flow through the command file. They provide for conditional testing, and branching. There is also a statement to invoke another command file.

Labels

A label consists of a line which begins with a colon (:) followed immediately with a label name. The label must be the first token on the line (i.e. it may be preceded with blanks). An example of a label is:

```
:error
```

Only the first 8 characters of a label name are used, although the name may be longer. The label may be followed by any statement.

If Statement

The conditional statement looks like:

IF (condition) statements

The condition is tested and if it is true, the statements on the remainder of the line are executed. Otherwise, the statements are skipped. A condition consists of compare operators, with AND and OR operators. Parentheses may be used to control the order of comparison.

The comparison operators are:

=	- Test for equality
<>	- Test for inequality
>	- Test for greater than
<	- Test for less than
>=	- Test for greater or equal
<=	- Test for less than or equal

Examples:

```
if (@A = ":") goto gotcolon;  
if (@1 = "send") goto dosend;  
:loop1 @a = getch; if (((@a <> ":") and (@a <> "?")) goto loop1;
```

Branching Statements

Several special statements are provided to control the order statements if the command file is invoked.

EXIT message;

If this statement is encountered, all processing of the command file is stopped and a return is made to the next highest level. The given message is shown on the current screen. The message argument is optional. The message may be a string or variable name.

CANCEL message;

If this statement is encountered, all processing of all command files is stopped and a return is made to the keyboard state. The given message is shown on the current screen. The message argument is optional. The message may be a string or variable name.

GOTO label;

This causes processing to begin at the line with the given label name.

Invoke Statement

You may begin execution in another command file with the Invoke statement, which looks like:

INVOKE filename [arg1 ... arg9];

This begins processing the given filename. When the end of file is reached or an EXIT statement is encountered in that file,

execution will continue in the current file at the statement following the INVOKE statement. INVOKEs may be nested up to 8 levels. If no extension is given for the filename, the extension .ACT is assumed.

After the filename, you may give up to 9 arguments to the file. These arguments may be referenced in the invoked file as variables @1 through @9. The argument may be a string, number, or variable.

Appendix B – Emulation File Format

An emulation file is broken into several sections. These sections must be present (even if they contain no statements) and they must be in the correct order. The sections are as follows:

INITIAL	The initialize section
INBUFFER	Definition of input buffer translations
INBUFFER2	Definition of secondary input buffer (if any)
OUTBUFFER	Definition of output translations
HOSTCODE	Definition of escape codes

Each of these sections begins with a line containing the above keyword on a line by itself, followed by statements, and followed by the keyword 'END' on a line by itself.

B.1 The INITIAL section

This section consists of call ACTION file statements. These statements are typically used to invoke functions, assign values to function keys, and assign values to builtin variables. Any ACTION function may be invoked. In addition, several special functions exist for the purposes of emulation. These functions are defined later in this document.

B.2 The INBUFFER section

The "input buffer" defines translations to take place when a character is read from the host port. This is useful if you want a character the host sends to have a different meaning than BitCom would normally apply to it. For example, when BitCom receives a FF (form feed) character, bit clears the screen. However, some terminals treat a FF like a LF (line feed) character. Thus, you would have a line like:

0x0C LF

The format of a line is: The hexadecimal representation of the character to be translated, followed by 1) a special mnemonic (e.g. LF) or 2) followed by another hexadecimal number (e.g. 0x1F).

The mnemonics understood by BitCom, and the action taken when that character is received from the host is as follows:

ESC	Begins an escape sequence
BEL	Speaker beeps
HT	Horizontal tab
LF	Line feed

BS	Destructive backspace
BS2	Non-Destructive backspace
FF	Clear screen
CR	Carriage return
DC1	XON character
DC3	XOFF character
CAN	Cancels escape sequence if one is being received
NUL	Ignored on input
SI	Switch active buffer to G0 (defined by z#setbuf)
SO	Switch active buffer to G1 (defined by z#setbuf)
ENQ	Transmit string defined by z#setenq0

At the beginning, the input buffer is initialized such that no translation will take place.

B.3 The INBUFFER2 Section

This section has the same format as the INBUFFER section, but defines a secondary input buffer. Some terminals have a second buffer which interprets characters differently (e.g. a special graphics symbol set).

B.4 The OUTBUFFER Section

This section has the same format as the INBUFFER section, but defines an output translation buffer. This buffer is used to translate any character sent to the host to a different character.

B.5 The HOSTCODE Section

This section defines the escape codes for the terminal.

The first token on the line defines the escape code. The code may not contain blanks, unless the blank is preceded by a backslash character (\). The escape code should NOT contain the ESC character in this definition. The special sequence \c (where "c" is any character) is taken to mean a numeric pattern will be found in this position.

The first non-blank after the first token begins an ACTION line that will be executed when the given escape code is detected. This action line may also contain the special sequence \c (where "c" is the same character as used in the first token). This sequence will be replaced with the number found when the pattern was matched.

This is all best explained with examples:

Pattern	Action
[N];\cf	poscur(\c,\l)
[6n	"\$1b[@row.:@col.R"

If the sequence "ESC[20;30f" were received, the function "poscur(30,20)" would be invoked. If the sequence "ESC[6n" were received, the string "ESC[20;30R" would be transmitted to the host (note that the builtin variables @col and @row are used for this).

There may also be a percent sign (%) followed by a letter in the pattern and action. In this case, the pattern means a single character will be in this position, and the "%c" in the action will be replaced by the numeric value of the character found.

B.6 Special Emulation Functions

The following functions may be called by any ACTION file, but are placed here since they are mostly useful to the terminal emulation function.

Cursor Movement

z#Curup(n,x)	Move cursor up 'n' lines. x=1 means scroll at top.
z#Curdn(n,x)	Move cursor down 'n' lines. x=1 means scroll at bottom.
z#Curfwd(n)	Move cursor right 'n' columns.
z#Curbak(n)	Move cursor left 'n' columns.

Save Environment

z#Savpos	Save current cursor position and attribute.
z#Respos	Restore cursor position and attribute (saved with z#Savpos).
z#Savorg	Save current origin flag.
z#Resorg	Restore origin flag (saved with z#Savorg).
z#Savbuf	Save current character buffer settings.
z#Resbuf	Restore buffer settings (saved with z#Savbuf).

Global Flags

z#Setatt(n)	Set current attribute. Typical values for 'n' are:
07	Normal
112	Reverse video
02	Green or normal
14	Bold or bright yellow
01	Underscore or blue
128	Turn on Blink

<code>z#Setbuf(n,m)</code>	<p><code>n=1</code> means SI will enable 1st input buffer.</p> <p><code>n=2</code> means SI will enable 2nd input buffer.</p> <p><code>n=3</code> means SO will enable 1st input buffer.</p> <p><code>n=4</code> means SO will enable 2nd input buffer.</p> <p>If <code>m=1</code>, active character set is changed immediately as if <code>n</code> SI character had been received.</p>
<code>z#Origin(n)</code>	Sets flag so cursor positions are taken to be relative to scroll window (<code>n=2</code>) or absolute (<code>n=1</code>)
<code>z#Awrap(n)</code>	Sets flag so characters displayed beyond end of screen will wrap (<code>n=0</code>) or truncate (<code>n=1</code>). <code>n=2</code> means wrap characters and wrap cursor movements.
<code>z#Setmar(r1,r2)</code>	Set scroll window to <code>r1</code> through <code>r2</code> .
<code>z#Lfnl(n)</code>	Sets flag so linefeeds (LF) are treated like newline (<code>n=1</code>) or not (<code>n=2</code>).
<code>z#Setenq(s)</code>	Sets message to be sent when an ENQ char received.
<code>z#Setflg(l,n)</code>	Sets a flag to <code>n</code> (a number). 'l' is a single character in the range 'a' - 'z'.
<code>z#Flg(l)</code>	Returns value of given flag.
<code>z#Setins(n)</code>	<code>n=0</code> specifies replace mode, <code>n=1</code> sets insert mode

Screen Clear

<code>z#Clreol</code>	Clear from cursor to end-of-line
<code>z#Clrbol</code>	Clear from begin-of-line to cursor
<code>z#Clrln</code>	Clear line containing cursor
<code>z#Clreos</code>	Clear from cursor to end-of-screen
<code>z#Clrbos</code>	Clear from begin-of-screen to cursor

Tab Setting

<code>z#Tabset</code>	Set tab mark at current cursor column.
<code>z#Tabclr</code>	Clear tab mark at current cursor column.
<code>z#Tabcla</code>	Clear all tabs.

Miscellaneous

z#Inslin(n)	Inserts n lines at cursor line
z#Dellin(n)	Deletes n lines at cursor line
z#Del(n)	Deletes character at cursor, and shifts line left
z#Chr(n)	Converts n to a single character and writes it to the port

APPENDIX C

VT100 AND IBM 3101 EMULATION KEY BOARD ASSIGNMENTS

Included in the diskette are 3 emulators: IBM 3101, VT100A, and VT100B. The function key assignments used on your PC are as follows:

VT100 KEYS

F1 - F4
Arrow keys
Delete
App-0 to App-9
App_-
App_
App ENTER

IBM PC KEYS

F1 - F4
Arrow keys
Del
Alt 0 to Alt 9 (Press Alt and 0)
Alt - (press Alt and -)
Alt =
Home

IBM 3101 KEYS

F1 to F8
Clear
Erase Input
Erase EOL
Erase EOS
Home
Arrow keys

IBM PC KEYS

F1 to F8
Alt c (Press Alt and c)
Alt i
Alt e
Alt s
Home
Arrow Keys

Appendix D – BitCom Error Messages

The following messages may be encountered during normal use of the BitCom system.

Break Detected

The host system to which you are connected with sent a “break” signal. No action is taken by BitCom in this case.

COMM.DAT: file invalid!

This error occurs when the COMM.DAT file has been corrupted. Always keep periodic backups of your COMM.DAT file so you may restore the file if this error occurs.

COMM.DAT: not found

The COMM.DAT file could not be read. This file is required for normal communications.

COMM.DAT: some data lost!

This error occurs usually when you have added a record and then suffered a power failure without exiting BitCom normally. The record(s) you added are probably missing, but previous data will be OK.

COMM.DAT: write error!

The COMM.DAT file could not be written for some reason (possibly out of disk space or you removed the disk). This file is written to during normal communications.

COMM.GLO: open error!

The COMM.GLO file could not be opened. This error causes termination of BitCom. The COMM.GLO file must be present in the current drive and directory.

COMM.GLO: write error!

The COMM.GLO file could not be written for some reason (possibly out of disk space). This file is only written to when you change the global BitCom parameters.

COMx: Not Functional

The indicated port number is not functional. You will get this if the serial port is not working for any reason. Check the switches

on your internal modem or your serial card. Make sure it is set to the correct communications port (e.g., COM1).

Cannot Dial/Answer with defaults

The default record (record 0) is only provided to specify default values for new records. It may not be used for any other purpose.

Cannot add while connected

You cannot add a new communications record when you are currently connected (though you can change the values of the current connection record).

Cannot delete record 0

The default record (record 0) is only provided to specify default values for new records. It may not be used for any other purpose.

Cannot delete while connected

You cannot delete a new communications record when you are currently connected (though you can change the values of the current connection record).

Connection ended

This is not usually an error, but is displayed when connection is ended by the remote host computer.

Data bits must be 8 for Xmodem

For Xmodem file transfers, you must have specified 8 data bits as a communications parameter.

Field full

You have tried to insert a character in a field, when the last character position of the field is full.

File=<filename> not found

The named file could not be found. Provide the correct filename and re-try.

Framing Error

A received character did not have a valid stop bit. If this error occurs frequently, try changing the number of stop bits.

Invalid hex digit in modem string

A \$xx sequence in a modem string (like the DIAL PREFIX) contains an invalid hex digit. Valid hex digits are 0-9 and A-F.

Key ignored, <send,receiv>ing file

If you are sending or receiving a file, any keys you press will be ignored (except for the "Cancel-File-Transmit" key).

No CTS Signal

The modem has not returned a "Clear to Send" signal. If you have an external modem, this error may occur if it is turned off.

No DSR Signal

The modem has not returned a "Data Set Ready" signal. If you have an external modem, this error may occur if it is turned off. If your serial card is not connected to a modem, this error may occur. Make sure you are talking to the right communications port (e.g., COM1 or COM2).

No memory for Select Menu

This may occur if you have an extremely large (more than 200 records) communications data file. BitCom will still operate normally, except you will not get the selection menu.

Not enough memory

This may occur if your system has less than 192K bytes of available memory. In most cases you may continue using BitCom, though some functions may be disabled.

Not special key...

In re-defining a special key, you pressed a key which is not a "special" key. You get this message if you press a normal letter or number key, instead of a function key, "alt" key, or "ctl" key.

Numbers only

You have tried to enter a non-numeric character in a field for which only numbers are valid.

Open failed for <filename>

The named file could not be opened. Provide the correct filename and re-try.

Overrun Error

The BitCom character receive buffer has overflowed. If the host computer sends characters faster than BitCom can type them, and the host does not understand the X-on/X-off protocol, you may get this error. This will normally only occur at high baud rates (greater than 2400 baud), since the BitCom buffer is 2048 bytes in size. Some received characters will be lost.

Parity Error

A received character does not have correct odd or even parity. Try changing the parity setting or number of data bits.

Synch. error, cannot continue

During Xmodem transfers, it is possible for the two computers to get out of sync. You should re-try the file transmit if you get this error, as it is likely caused by data line noise.

Time Out

During communications, a device timeout occurred. This is usually caused by a modem failure.

Unknown function

You have entered a letter or number which is not known as a menu option.

Waiting for X-ON

When you are sending a text file in ASCII mode, you may receive this message if the computer you are sending the file to cannot receive the file as quickly as BitCom is sending it. The message will go away as soon as the host computer gives the go ahead.

APPENDIX E

Script File Error Messages.

The following errors will only be seen if you are writing either a SCRIPT file or an ACTION file.

Already connected

The SCRIPT file invoked the 'docomm' function to begin communications, but the line is already connected.

Builtin variable not set-able

An attempt was made to assign a value to a 'read-only' builtin variable. Certain builtin variables may not be assigned to (e.g. @connect).

Cannot 'select' while connected

The SCRIPT file called the 'select' function, but you are still connected. Use the 'hangup' function to disconnect the line before doing a 'select'.

Cannot assign

An attempt was made to assign a value to something other than a variable.

EOF arg only valid for "ascii" mode

The 'recvfile' function was invoked with mode="xmodem". For this mode, it is not necessary to specify an EOF string.

Expected comma

BitCom expected a comma to be present, but found something else. Commas are usually used to separate arguments to a function.

Expected compare operator

Within an IF condition, BitCom expected a compare operator (e.g. =, >) but found something else.

Expected label name

A label name was not given following a GOTO statement. The label name after the GOTO must include the colon (e.g. 'goto :top').

Expected label or statement

BitCom expected a statement or label, but found something else. Consult help or the manual for valid statement syntax.

Expected left paren

BitCom expected a left parentheses to be present, but found something else. Parenthesis are usually present after function names and the 'if' statement.

Expected right paren

BitCom expected a right parenthesis to be present, but found something else.

Expected right paren or and-or

BitCom expected a right parenthesis or the keyword AND or OR within the IF condition, but found something else.

Expected semi-colon or end-of-line

BitCom reached the logical end of a statement, but found something other than a semicolon (which can separate statements) or the end of line.

Expected variable or constant

BitCom expected a variable name, constant value, or function invocation, but found something else.

File transfer already active!

The 'recvfile' or 'sendfile' function was called, but a file is currently being transferred.

File=<filename> 'goto' error

A "goto" within a SCRIPT or ACTION file specified a non-existent label. Either add the label or correct the "goto" and re-try.

Invalid \$xx hex character

A character may be specified as a hexadecimal value in a d-string by giving a '\$' followed by exactly 2 hex digits. Hex digits may have a value of 0-9 and A-F.

Invalid <builtin-variable-name> setting

An attempt was made to specify an invalid value to a builtin variable. Consult help or manual for valid settings.

Invalid Cwait argument

A null string ("") was given as an argument to the 'cwait' function.

Invalid EOF string

A null string ("") was given as an EOF string to the 'recvfile' function.

Invalid file transfer mode

The file transfer mode given in a 'sendfile' or 'recvfile' function call is invalid. It must be given as a string, and may be either "ascii" or "xmodem".

Invalid number

BitCom expected a numeric value, but found something else.

Invalid time units

The 'twait' function was given an invalid string for the time units. Valid values are "hsec", "sec", "min" and "hour".

Invalid time value

The 'wait' function was given a non-valid time (i.e. hour was not in the range 0-23, or minutes was not in the range 1-60).

Label only valid in a file

A label was given in a function key definition string. Labels are only valid in SCRIPT or ACTION files.

Misplaced keyword

A special keyword (e.g. IF, EXIT, INVOKE, etc.) was found in an inappropriate location.

Must use "select" function first

An attempt was made to set or query the value of a builtin variable, but no record has been made current either with the 'select' function or through BitCom menus.

Reached EOL in "d-string"

A "d-string" must begin with a double quote and end with a double quote, and must be given on a single input line.

Reached EOL in 's-string'

An 's-string' must begin with a single quote and end with a single quote, and must be given on a single input line.

Reached end-of-line in comment

A comment must begin with a '[' and end with a ']', and both the opening and closing braces must be given on a single input line.

Run file cancelled

The break key was pressed while a SCRIPT or ACTION file was running. This is an informative message only.

Too many INVOKE levels

You may nest "invoke" statements (that is, an "invoke" file may invoke yet another file), but only up to 8 levels deep.

Unknown label name

The label name given following a GOTO statement could not be found in the current SCRIPT or ACTION file.

Unknown or misplaced character

A character unknown to BitCom was found in an Action or Script file. You may need to surround the character with quotes.

Unknown variable name

A variable name was given (a string beginning with an at-sign '@') which is unknown to BitCom. Refer to the help or documentation for a list of valid variables.

Use 'select' function first

The SCRIPT file invoked 'docomm' before specifying the record to use for communications with the 'select' function.

APPENDIX F

List Of Files On BitCom Diskette.

BBSGET	ACT	546	9-06-84	3:48p
BBSRECV	ACT	308	9-02-84	2:01 p
BIGFOOT	ACT	353	10-08-84	11:04p
COMSERV	ACT	219	10-08-84	11:06p
HANDYKEY	ACT	264	9-06-84	3:15p
PUBDOM	ACT	476	10-08-84	11:06p
RCVSVCS	ACT	323	9-04-84	12:04a
RECVSVCS	ACT	313	9-03-84	10:31p
SVCS	ACT	339	10-08-84	11:08p
SVCSGET	ACT	323	9-04-84	12:04a
COMM	DAT	4929	1-01-80	12:43a
3101	EMU	1005	10-16-84	9:16a
VT100A	EMU	3628	1-28-85	4:17p
VT100B	EMU	4053	4-05-85	10:39a
COMM	GLO	272	4-29-85	10:43a
COMM	HLP	2163	9-05-84	5:18p
FXFR	HLP	923	9-11-84	9:30a
INIT	HLP	5505	2-14-85	9:04p
MAIN	HLP	1760	9-05-84	4:52p
MENC	HLP	1948	9-11-84	9:29a
PARM	HLP	10207	3-19-85	9:28p
SCRIP	HLP	18032	1-29-85	7:00a
SELE	HLP	5304	10-04-84	10:18p
DIAL	SCP	919	9-11-84	10:00p
EASYLINK	ACT	107	1-01-80	12:06a
ERRORS1	DOC	5535	1-30-85	9:52p
ERRORS2	DOC	5460	1-30-85	9:52p
REL2P2	DOC	4138	2-24-85	7:21a
REL2P3	DOC	4357	3-21-85	12:35a
BITCOM	EXE	83200	4-05-85	10:28a

APPENDIX G

ASCII Code Table.

Decimal Value	Control Code	ASCII Code	Decimal Value	ASCII Code	Decimal Value	ASCII Code
0		NUL	48	0	96	
1	CTRL A	SOH	49	1	97	a
2	CTRL B	STX	50	2	98	b
3	CTRL C	ETX	51	3	99	c
4	CTRL D	EOT	52	4	100	d
5	CTRL E	ENQ	53	5	101	e
6	CTRL F	ACK	54	6	102	f
7	CTRL G	BEL	55	7	103	g
8	CTRL H	BS	56	8	104	h
9	CTRL I	HT	57	9	105	i
10	CTRL J	LF	58	:	106	j
11	CTRL K	VT	59	;	107	k
12	CTRL L	FF	60	<	108	l
13	CTRL M	CR	61	=	109	m
14	CTRL N	SO	62	>	110	n
15	CTRL O	SI	63	?	111	o
16	CTRL P	DLE	64	@	112	p
17	CTRL Q	DC1	65	A	113	q
18	CTRL R	DC2	66	B	114	r
19	CTRL S	DC3	67	C	115	s
20	CTRL T	DC4	68	D	116	t
21	CTRL U	NAK	69	E	117	u
22	CTRL V	SYN	70	F	118	v
23	CTRL W	EOB	71	G	119	w
24	CTRL X	CAN	72	H	120	x
25	CTRL Y	EOM	73	I	121	y
26	CTRL Z	SUB	74	J	122	z
27	CTRL [ESC	75	K	123	{
28	CTRL \	FS	76	L	124	
29	CTRL]	GS	77	M	125	}
30	CTRL ^	RS	78	N	126	~
31	CTRL _	US	79	O	127	DEL
32		SP	80	P		
33		!	81	Q		
34		"	82	R		
35		#	83	S		
36		\$	84	T		
37		%	85	U		
38		&	86	V		
39		'	87	W		
40		(88	X		
41)	89	Y		
42		*	90	Z		
43		+	91	[
44		,	92	\		
45		-	93]		
46		.	94	^		
47		/	95	_		

BitCom Release 2.3 Update Notice

This release of BitCom corrects the following problems in release 2.0:

- 1) If the file COMMAND.COM is not present when trying to issue DOS commands, the message 'COMMAND.COM not found' will be given.
- 2) X-on and X-off will only affect BitCom when it is sending a file in ASCII mode. Previously, it was possible to get in a state where all keys were ignored because an X-off character had been received.
- 3) Changing the communications setting of "Echo" (also known as Full/Half Duplex) will no longer cause the line to be dropped and reconnected.
- 4) Errors in the "VT100A.EMU" and "VT100B.EMU" emulation files have been corrected. In addition, the function key mapping has been CHANGED to the following:

VT100 Numeric Keypad IBM-PC Keypad (Numeric Shift)

NOTE:

Since the IBM-PC keypad does not have a comma, the function of that key has been re-assigned to key F10. These key assignments can easily be changed by editing the file VT100A.EMU or VT100B.EMU.

- 5) Serial printers now work as documented.

The following features have been added:

- 1) It is now possible to define function values for the keys "Ctl-C", "Ctl-H", "Ctl-I", "Ctl-J", "Ctl-M", and "Ctl-P". Previously these were omitted because of conflicts with other keys.
- 2) The following keys may now be defined:

@Ctlret	- Control Return
@NL-0	- Numeric Keypad 0
@NL-1	- Numeric Keypad 1
@NL-2	- Numeric Keypad 2
@NL-3	- Numeric Keypad 3

@NL-4	- Numeric Keypad 4
@NL-5	- Numeric Keypad 5
@NL-6	- Numeric Keypad 6
@NL-7	- Numeric Keypad 7
@NL-8	- Numeric Keypad 8
@NL-9	- Numeric Keypad 9
@NL-Dsh	- Numeric Keypad -
@NL-Pls	- Numeric Keypad +
@NL-Dot	- Numeric Keypad .

- 3) The SCRIPT/ACTION function "substr" has been added. This allows the creation of sub-strings. One possible use of this feature would be a SCRIPT file which defines the capture file name to be some combination of the current date. Use as follows:

substr(string, start, len)

Returns a sub-string of the given string, starting at character 'start' (where 1 is the first character). The string will be 'len' characters long. The 'len' argument may be omitted, in which case the sub-string will be from 'start' till the end of 'string'.

- 4) Menu selection forms now display a hint message at the top stating which key to press to select a new value in a "selection" field.
- 5) When giving commands to DOS (either when in DOS mode, or when calling the "dos" SCRIPT/ACTION function), you may specify more than one command to be run by separating commands with the character '#'. For example, the function call 'dos("dir a:#dir b:")' lists the directories of both the A: and B: disks. If a command requires a '#', specify two '#'s in a row.
- 6) If the Num Lock or Caps Lock keys are pressed, the BitCom status line will show a reverse video NUM and CAPS.
- 7) It is now possible to specify ports COM1 through COM4 as the communications port.
- 8) It is now possible to give an arbitrary character to wait for from the host during ascii file transmittal. Do this in the Communications menu, option Lchar.